

Practical Fuzzy-Logic Design

The Fuzzy-Logic Advantage

In this article, Constantin shows us a situation where fuzzy logic solves a problem that traditional logic cannot. His intention is not to replace conventional techniques, however, just extend them.

FEATURE ARTICLE

Constantin von Altrock



Over the past few years, systems designers have heard some rather controversial discussion about a new design technology called fuzzy logic. Some disciples of fuzzy logic have pitched it as the magic bullet for every type of engineering problem, and they've made wild and unsupported claims.

Such behavior raises skepticism among control-engineering specialists, causing some to completely condemn fuzzy logic as a marketing fad of no technical value.

The discussion in Asia and Europe about fuzzy logic has been much less religious and has kept closer to its real benefits. Fuzzy logic is considered one of the many tools necessary to build systems solutions in a fast, cost-effective, and transparent fashion [1]. It's not more or less than that.

Previous *INK* articles have generated explicit questions in how fuzzy logic achieves superior performance compared to other engineering design techniques. My goal is to show you exactly that.

To illustrate how fuzzy logic solves real-world problems and to compare it head-to-head with conventional design techniques, I'll use the case study of the antisway crane controller shown in Photo 1. I selected this real-world application for a number of reasons.

Unlike other technical real-world applications, the works of a crane controller can be understood quickly.

Antisway control of a crane is a simple problem, but conventional engineering techniques have a hard time arriving at a practical solution. In contrast, fuzzy logic rapidly delivers a good solution which has been successfully implemented on many crane controllers.

In this article, I'll assume you're familiar with the concept of fuzzy logic. If not, refer to this issue's "An Introduction to Fuzzy Logic" by Chris Sakkas, to the last fuzzy-logic issue (*INK* 56), or to *Fuzzy Logic and Neuro-Fuzzy Applications Explained* [2].

SINGLE-VARIABLE CONTROL

Before I start with the case study, let me clarify some control-engineering basics. Closed-loop control mostly involves keeping a single figure con-



Photo 1—When transporting concrete modules for bridges and tunnels by a crane, sway of the load must be controlled. A fuzzy-logic controller uses the experience of a human crane operator. This crane has two heads, each capable of transporting 64 tons, and it employs an antisway controller based on fuzzy logic.

the mass flow of a liquid, or the temperature of a room.

To keep room temperature constant, a sensor delivers the value of the actual room temperature. The difference between this temperature and the desired room temperature is the temperature error.

The controller's objective is to get the temperature error to zero. To do this, it may turn a heater or air conditioner on or off. Such controllers are referred to as on/off-type controllers.

If the command variable of the process is continuous, such as with a gradual heat control, a so-called proportional-type (P) controller is used. This implements a control strategy that computes its output—the command variable of the process—by multiplying the error by a constant factor.

Many real-world processes involve a time lag. When controlling room temperature and the heater turns on, it takes a while before the heat reaches the sensor.

Because the controller gets the measured room-temperature variable with a time delay, it can overreact and apply more heat than what's required to get the temperature error to zero. This overreaction can cause unwanted oscillation of the room temperature.

To compensate for this effect, proportional-differential (PD) controllers

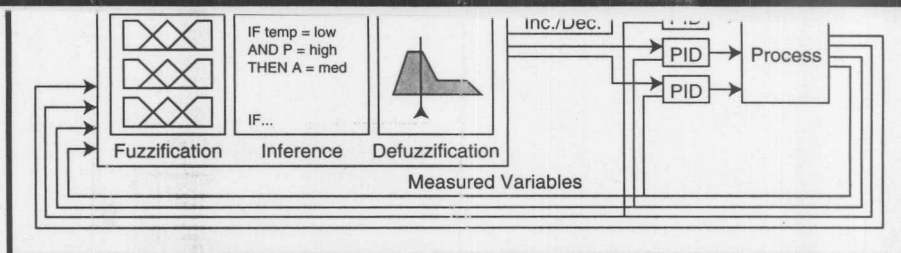


Figure 1—In many applications, keeping a single figure of the process constant is relatively easy and can be facilitated by conventional PID controllers. However, to optimize the operation point of the process, a supervisory controller modifies the set points of the PID controllers.

add the time derivative of the error multiplied by a constant to the error.

Proportional-integral-differential (PID) controllers also consider the integral of the error and add it to the output signal. This method ensures that the error eventually reaches zero because even a small offset in the error value is compensated for as its integral raises to a high value over time.

Because conventional controllers only use the error or signals derived from the error as input and the command variable of the process as output, they are called single-loop controllers. These controllers are usually simple to program and easy to tune.

Replacing such controller types by a fuzzy-logic controller only delivers a better performance in cases where the conventional controller doesn't cope well with the nonlinearities of the process under control [2, p. 82].

MULTIVARIABLE CONTROL

A much higher potential for fuzzy logic lies in the fact that it also facilitates the design of multivariable control systems. In a multivariable control loop, the control strategy considers not only one input variable (e.g., error) but different types of measured variables.

In the case of the room thermostat, a multivariable control strategy considers temperature error and room air humidity to set the command variables of the heater and air conditioner.

To deal with multiple input variables, a controller must assume a mathematical model of the controlled process. For the multivariable room-thermostat control strategy, such a model can be the humidity and temperature relationship which describes comfortable conditions.

In most real-world applications, the mathematical relation between the different variables cannot be described so easily. So, multivariable control is not commonly used.

However, application areas exist where multivariable-control strategies must be used, such as continuous process control as found in the food and chemical industries. Keeping single figures of the processes constant is easy in most cases. Controlling the plant's optimal operation point involves multiple variables.

Because of the difficulties involved with deriving mathematical models, automated control is rare and human operators often adjust the set points of the individual PID controllers.

This type of multivariable control is also referred to as supervisory control because the multivariable control strategy analyzes and supervises the set points of underlying control loops.

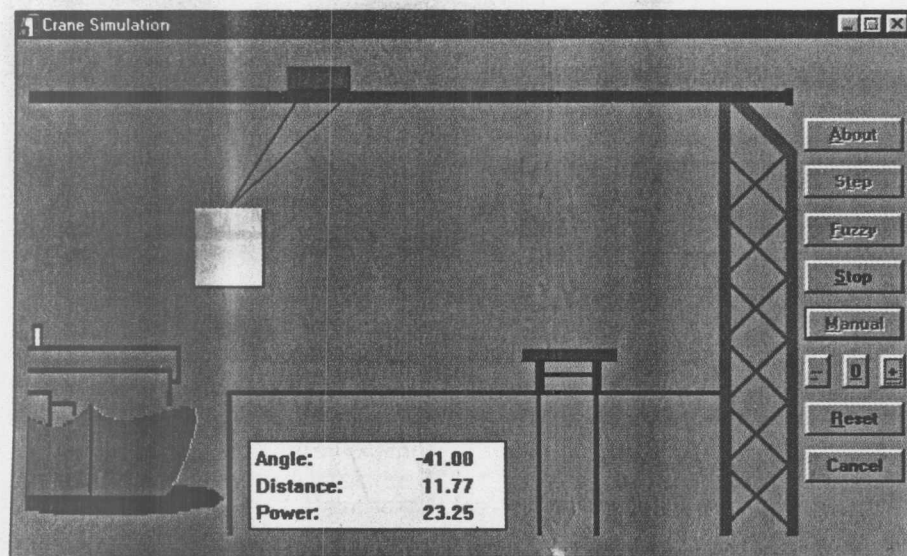


Photo 2—The software simulation of container-crane operation visualizes the operation of the fuzzy-logic antisway controller. The fuzzy-logic control strategy is to get the container from the ship to the target as quickly as possible. When the target is reached, the sway must be compensated before releasing the container.

BIG THINGS COME DOMINOTM Microcontroller



Starting at
\$99

Micromint's Domino-52 microcontroller is a "supercomputer" in less than 0.75 cubic inches. We've packed the most essential elements into one tiny package. Domino is a plug-and-go module, just attach +5V and a terminal or network. A simple keyed sequence saves an autostarting program in nonvolatile memory.

SPECIAL FEATURES

- 80C52 with ROM-resident, full floating-point BASIC
- 32K bytes SRAM and 32K bytes EEPROM
- Two PWM outputs, I²C bus
- Serial I/O: (up to 19,200 bps) RS-422, RS-485 & RS-232A
- Two interrupts and three timers
- Parallel I/O: 12 bits, 3 shared with ADC and I²C
- Power: +5V @ 15 mA;
Size: 1.75" x 1.062" x 0.4" potted
- A/D converter: 2 channels, 12 bits, 10k samples/sec.
- Connections: via 2x10, 0.1" dual-row header
- -20°C to 75°C operating temperature
- Industrial temperature available



MICROMINT, INC.

4 Park Street • Vernon, CT 06066

Call 1-800-635-3355

(860) 871-6170 • Fax (860) 872-2204

#119

Spreadsheet Rule Editor - RB1				
	IF		THEN	
	Angle	Distance	DoS	Power
1	zero	far	1.00	pos_medium
2	neg_small	far	1.00	pos_big
3	neg_big	far	1.00	pos_medium
4	neg_small	medium	1.00	neg_medium
5	pos_small	close	1.00	pos_medium
6	zero	zero	1.00	zero

Photo 3—Each row corresponds to a fuzzy rule. The columns Angle and Distance underneath IF form the conditions of the rules. Underneath THEN, the column Power forms the conclusion of the rules. The DoS (Degree of Support) column can contain an individual weight of each rule.

In contrast to conventional design techniques, fuzzy logic enables the design of such multivariable control strategies directly from human-operator experience or experimental results (see Figure 1).

Using such existing knowledge and circumventing the effort of rigorous mathematical modeling, the fuzzy-logic design approach delivers efficient solutions faster.

The question therefore becomes, "How can operator experience be put into a fuzzy-logic system?" The following case study of a container-crane antisway control illustrates this.

CONTAINER-CRANE CONTROL

Container cranes load and unload containers to and from ships in most harbors (see Photo 2). They pick up single containers with flexible cables mounted at the crane head.

The crane head moves on a horizontal track. When a container is picked up and the crane head starts to move, the container begins to sway. While sway does not affect the transport, a swaying container can't be released.

There are two trivial solutions to this problem. One is to position the crane head exactly over the target position and wait until the sway

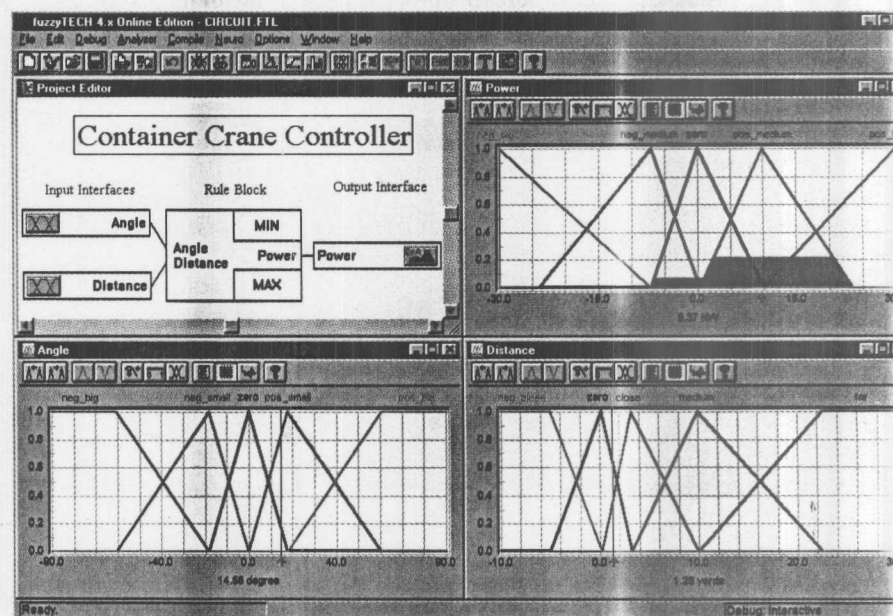


Photo 4—In this design of the fuzzy-logic antisway controller in fuzzyTECH, the upper left window shows the structure of the system with two inputs, one output, and one rule block. The upper right window visualizes the defuzzification of Power, and the lower windows show the fuzzification of Angle and Distance.

to see how the container sways. the control fuzzy-logic table in Photo 3. logic table in Photo the control strategy. the control strategy

Circuit Cellar INK®

Issue #75 October 1996

Issue #75 October 1996

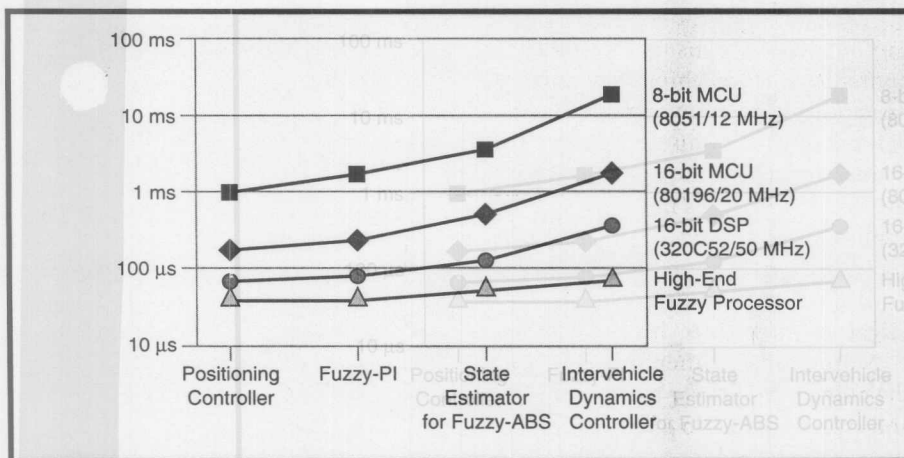


Figure 3—Using a set of standardized benchmark systems, the performance of different hardware platforms can be compared. On a 8051 MCU, a small fuzzy-logic system can be computed in about a millisecond. Faster microcontrollers and DSP can compute even very large fuzzy-logic systems in fractions of a millisecond.

- create an initial fuzzy-logic rule base—On a standard 12-MHz 8051 MCU, using all available knowledge on a small fuzzy-logic system compute in how the system should perform. just one millisecond, and 16-bit MCUs
- debug, test, and verify the system—can compute large fuzzy-logic systems offline for completeness and non- in the same amount of time. This ambiguity—use a software simulation or sample data of the process if it exists in this step. speed enables the integration of a fuzzy-logic system with most embedded system designs. However, some em-
- debug online—connect the fuzzy-—connected system applications require logic system to the process under to the even faster computation (e.g., antilock control and analyze its performance brakes in cars, ignition control, or hard-drive positioning). To expedite the fuzzy-logic algorithm, some semiconductor manufacturers include specific fuzzy-logic instructions with their new-generation MCUs. For example, Motorola's

Most systems today are developed using software tools like the one in Photo 4. Such development tools not only support all listed design steps, but they also can generate the entire system for different hardware platforms. For microcontrollers (MCUs), the tools generate assembly code. For PLCs, they generate function blocks. For PC/workstations, they generate C code.

Many years ago, the computation of the fuzzy-logic algorithm was so inefficient on a standard MCU that dedicated fuzzy-logic processors were developed. Today, code efficiency has improved dramatically. Rather, fuzzy logic extends the way automated control techniques are used in practical applications by adding supervisory control capabilities. The container-crane example demonstrates that fuzzy logic delivers a transparent, simple solution for a problem that's much harder to solve using conventional engineering techniques.

Figure 3 shows different MCUs, a DSP, and today's fastest fuzzy-logic processor computational performance for four benchmark systems. The time shown on the vertical logarithmic scale is the total time required to compute the complete fuzzy-logic system.

Constantin von Altrock began research on fuzzy logic with Hewlett-Packard in 1984. In 1989, he founded

and still manages the Fuzzy Technologies Division of Inform Software, a market leader in fuzzy-logic development tools and turn-key applications. You may reach Constantin at cva@inform-ac.com

SOFTWARE

You may download the complete animated software simulation of the container crane for Windows from Circuit Cellar's BBS together with a simulation-only version of fuzzyTECH and the FTL source code of the crane controller. You may download the source code of the benchmark suite from fuzzyTECH's Web site (<http://www.inform-ac.com>).

REFERENCES

- [1] C. von Altrock, "Fuzzy Logic Applications in Europe," in J. Yen, R. Langari, and L.A. Zadeh (eds.) *Industrial Applications of Fuzzy Logic and Intelligent Systems*, IEEE Press, Chicago, 275-310, 1995.
- [2] C. von Altrock, *Fuzzy Logic and NeuroFuzzy Applications Explained*, Prentice Hall, Englewood Cliffs, NJ, 1995.

SOURCES

68HC12 MCU
MCU Information Line
Motorola
P.O. Box 13026
Austin, TX 78711-3026
(512) 328-2268
Fax: (512) 891-4465
<http://freeware.aus.sps.mot.com/>

fuzzyTECH
Inform Software
2001 Midwest Rd.
Oak Brook, IL 60521
(630) 268-7550
Fax: (630) 268-7554
<http://www.inform-ac.com/>

IRS

404 Very Useful
405 Moderately Useful
406 Not Useful